

acmqueue Interactive Dynamics for Visual Analysis

A taxonomy of tools that support the fluent and flexible use of visualizations

Jeffrey Heer, Stanford University

Ben Shneiderman, University of Maryland, College Park

The increasing scale and availability of digital data provides an extraordinary resource for informing public policy, scientific discovery, business strategy, and even our personal lives. To get the most out of such data, however, users must be able to make sense of it: to pursue questions, uncover patterns of interest, and identify (and potentially correct) errors. In concert with data-management systems and statistical algorithms, analysis requires contextualized human judgments regarding the domain-specific significance of the clusters, trends, and outliers discovered in data.

Visualization provides a powerful means of making sense of data. By mapping data attributes to visual properties such as position, size, shape, and color, visualization designers leverage perceptual skills to help users discern and interpret patterns within data.¹¹ A single image, however, typically provides answers to, at best, a handful of questions. Instead, visual analysis typically progresses in an iterative process of view creation, exploration, and refinement. Meaningful analysis consists of repeated explorations as users develop insights about significant relationships, domain-specific contextual influences, and causal patterns. Confusing widgets, complex dialog boxes, hidden operations, incomprehensible displays, or slow response times can limit the range and depth of topics considered and may curtail thorough deliberation and introduce errors. To be most effective, visual analytics tools must support the fluent and flexible use of visualizations at rates resonant with the pace of human thought.

The goal of this article is to assist designers, researchers, professional analysts, procurement officers, educators, and students in evaluating and creating visual analysis tools. We present a taxonomy of interactive dynamics that contribute to successful analytic dialogues. The taxonomy

TABLE 1: Taxonomy of interactive dynamics for visual analysis

Data & View Specification	<p>Visualize data by choosing visual encodings.</p> <p>Filter out data to focus on relevant items.</p> <p>Sort items to expose patterns.</p> <p>Derive values or models from source data.</p>
View Manipulation	<p>Select items to highlight, filter, or manipulate them.</p> <p>Navigate to examine high-level patterns and low-level detail.</p> <p>Coordinate views for linked, multi-dimensional exploration.</p> <p>Organize multiple windows and workspaces.</p>
Process & Provenance	<p>Record analysis histories for revisitation, review and sharing.</p> <p>Annotate patterns to document findings.</p> <p>Share views and annotations to enable collaboration.</p> <p>Guide users through analysis tasks or stories.</p>

consists of 12 task types grouped into three high-level categories, as shown in table 1: (1) data and view specification (visualize, filter, sort, and derive); (2) view manipulation (select, navigate, coordinate, and organize); and (3) analysis process and provenance (record, annotate, share, and guide). These categories incorporate the critical tasks that enable iterative visual analysis, including visualization creation, interactive querying, multiview coordination, history, and collaboration. Validating and evolving this taxonomy is a community project that proceeds through feedback, critique, and refinement.

Our focus on interactive elements presumes a basic familiarity with visualization design. The merits and frailties of bar charts, scatter plots, timelines, and node-link diagrams, and of the visual-encoding decisions that underlie such graphics, are certainly a central concern, but we will largely pass over them here. A number of articles and books address these topics in great detail,^{11,12,16,52} and we recommend them to interested readers.

Within each branch of the taxonomy presented here, we describe example systems that exhibit useful interaction techniques. To be clear, these examples do not constitute an exhaustive survey; rather, each is intended to convey the nature and diversity of interactive operations. Throughout the article the term *analyst* refers to someone who uses visual analysis tools and not to a specific person or role. Our notion of analyst encompasses anyone seeking to understand data: traditional analysts investigating financial markets or terrorist networks, scientists uncovering new insights, journalists piecing together a story, and people tracking various facets of their lives, including blood pressure, money spent, electricity used, or miles traveled.

DATA AND VIEW SPECIFICATION

To enable analysts to explore large data sets involving varied data types (e.g., multivariate, geospatial, textual, temporal, networked), flexible visual analysis tools must provide appropriate controls for specifying the data and views of interest. These controls enable analysts to selectively *visualize* the data, to *filter* out unrelated information to focus on relevant items, and to *sort* information to expose patterns. Analysts also need to *derive* new data from the input data, such as normalized values, statistical summaries, and aggregates.

VISUALIZE

Perhaps the most fundamental operation in visual analysis is to specify a visualization of data: analysts must indicate which data is to be shown and how it should be depicted. Historically, this process required custom programming of a specific visualization component. Within user interfaces, such visualization “widgets” are often presented in a *chart typology*, a palette of available visualization templates (bar charts, scatter plots, map views, etc.) into which analysts can slot their data. This method of interaction will be immediately familiar to users of spreadsheet programs: users select a chart type and assign data variables to visual aspects such as the X/Y axes and the size or color of visualized marks. A chart typology has the benefits of simplicity and familiarity, but it also limits the types of possible visualizations and makes it cumbersome to try out different visualizations of the same data.

Some visualization system designers have explored alternative approaches. Classic scientific visualization systems¹ and more recent platforms for artistic expression⁹ use *data-flow graphs*, in which the visualization process is deconstructed into a set of finer-grained operators for data import, transformation, layout, coloring, etc. Analysts interactively chain these operators together

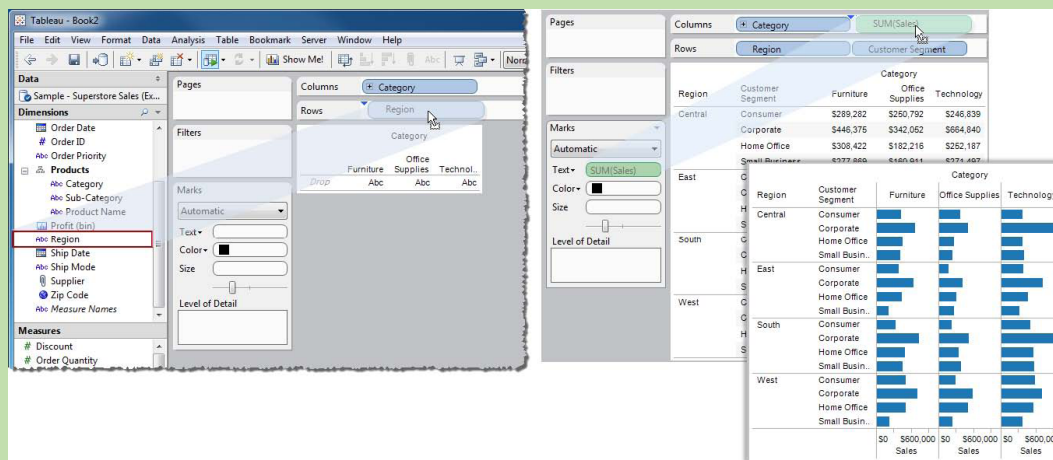
to construct novel displays. Through flexible combinations of operators, data-flow models can enable a larger space of visualization designs. Data-flow systems require more input effort than chart typologies, however, and may be limited by the set of available operators. In many cases, novel designs require analysts with programming expertise to develop new building blocks for the system.

Other systems are based on *formal grammars* for visualization construction. These grammars constitute high-level languages for succinctly describing how data should be mapped to visual features. By combining a handful of such statements, analysts can construct complex, customized visualizations with a high degree of design control. This approach is used by a number of popular data visualization frameworks such as Leland Wilkinson's *Grammar of Graphics*,⁵⁷ ggplot2 for the R statistical analysis platform,⁵⁶ and Protovis for HTML5.¹⁰ Each of these requires at least minimal programming ability, however.

Tableau⁵¹ (née Polaris⁵⁰) provides an example of visualization specification by drag-and-drop operations: analysts place data variables on “shelves” corresponding to visual encodings such as spatial position, size, shape, and color (see figure 1). The visual specification is then translated into an underlying formal grammar that determines both the visualization design and corresponding queries to a database. This approach leverages the expressiveness of formal grammars while avoiding the need for programming. Another advantage is that formal grammars can be augmented with automated design facilities: a system can generate multiple visualization suggestions from a partial specification.^{37,38,44} While systems based on formal grammars are both fluent and expressive, users need to understand the underlying generative model, which imposes a steeper learning curve than the more familiar chart typology.

FIGURE 1

Visual encoding via drag-and-drop actions in Tableau⁵¹



Fortunately, these methods are not mutually exclusive. Analysts can apply a data-flow system or formal grammar to define new components to include within a chart typology, leveraging the improved expressiveness of the former and the ease of use of the latter. Novel interfaces for visualization specification are still needed. A formal grammar that uses graphical marks (rectangles, lines, plotting symbols, etc.) as its basic primitives provides a conceptual model compatible with interactive design tools. New tools requiring little to no programming might place custom visualization design in the hands of a broader audience.

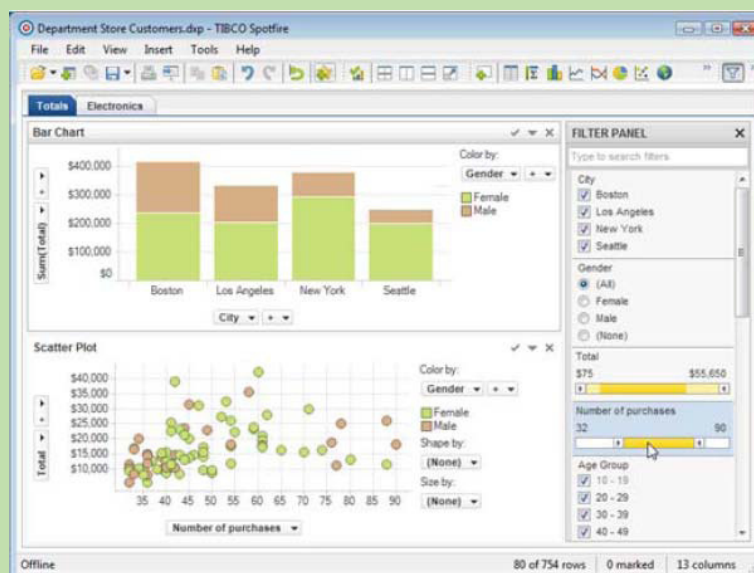
FILTER

Filtering of data values is intrinsic to the visualization process, as analysts rarely visualize the entirety of a data set at once. Instead, they construct a variety of visualizations for selected data dimensions. Given an overview of selected dimensions, analysts then often want to shift their focus among different data subsets—for example, to examine different time slices or isolate specific categories of values.

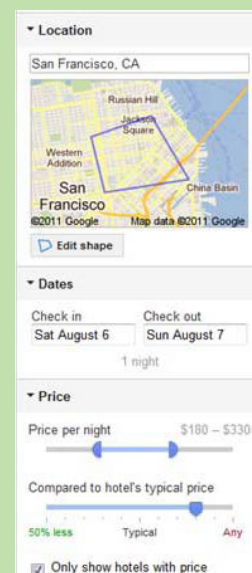
Designers have devised a variety of interaction techniques to limit the number of items in a display. Analysts might directly select (e.g., “lasso”) items in a display and then highlight or exclude them; we discuss these forms of direct view manipulation later. Another option is to use a suite of auxiliary controls, or *dynamic query widgets*,⁴⁷ for controlling item visibility (see figures 2, 3, and 4). The choice of appropriate widget is largely determined by the underlying data type. Categorical or ordinal data can be filtered using simple radio buttons or checkboxes (when the number of distinct items is small), or scrollable lists, hierarchies, and search boxes with

FIGURE 2

Examples of dynamic query filter widgets



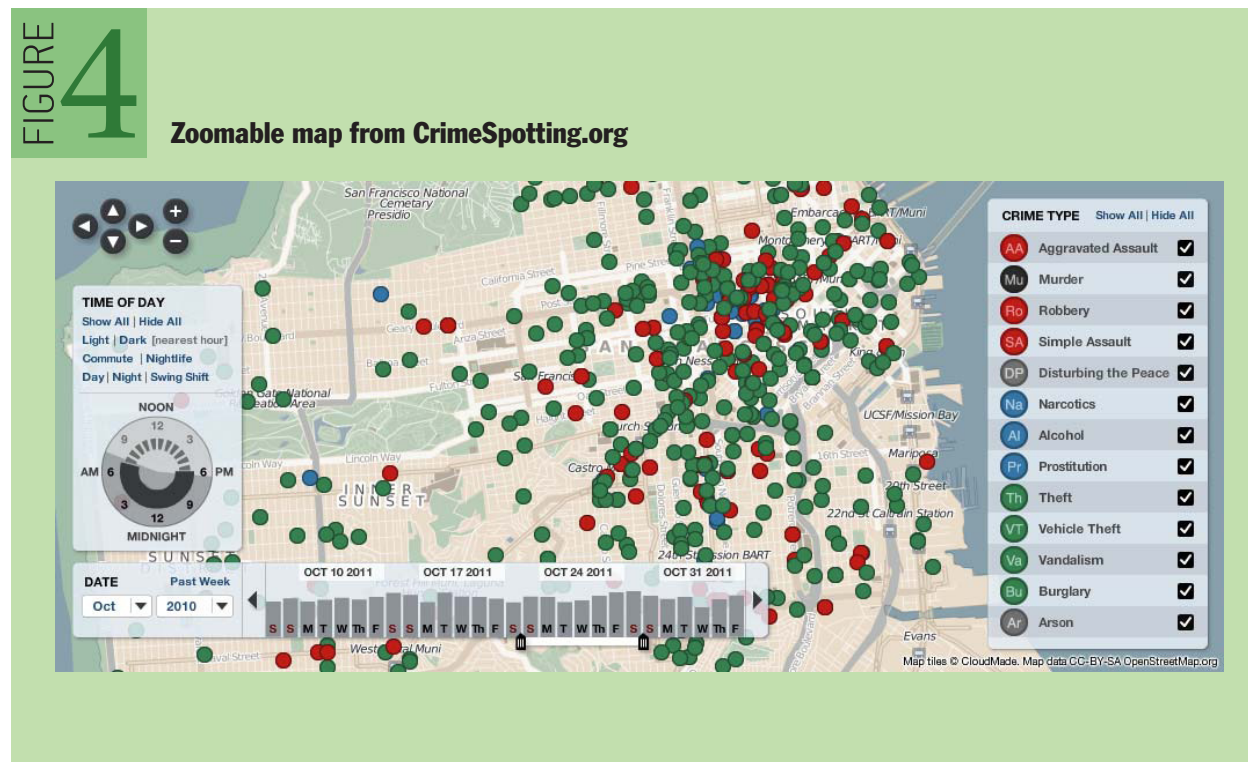
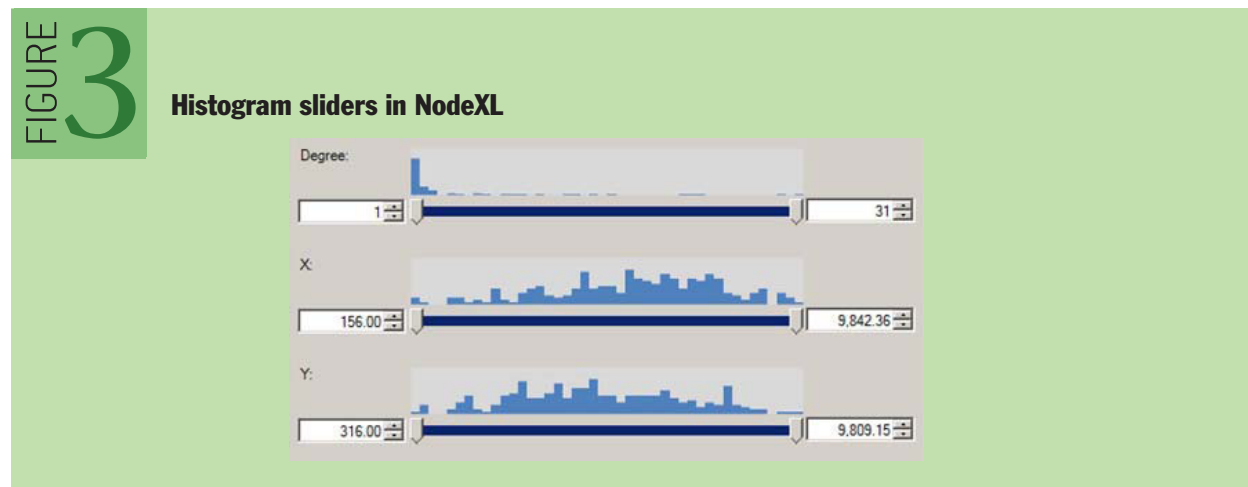
Spotfire



Google Hotel Search

autocomplete (when the number of distinct items is large or contains arbitrary text). Ordinal, quantitative, and temporal data can also be filtered using a standard slider (for a single threshold value) or a range slider (for specifying multiple endpoints). When coupled with realtime updates to the visualization, these widgets allow rapid and reversible exploration of data subsets. In figure 2, Spotfire (left) provides a variety of controls for filtering visualized data: checkboxes and radio buttons filter categorical variables, while range sliders filter numerical values; on the right, Google Hotel Search provides widgets for geographic, date, and price ranges. Query controls can be further augmented with visualizations of their own: figure 3 shows a range slider augmented with a histogram of underlying values.

Expert analysts also benefit from more advanced functionality. For example, a search box might



support sophisticated query mechanisms, ranging in complexity from simple keyword search, to regular expression matching, to a full-fledged structured query language. While these additional mechanisms may not support rapid, incremental exploration as fluently as graphical widgets, they provide a means for expressing more nuanced criteria. Filtering also interacts with other operations: filtering widgets may operate over data sorted in a user-specified manner (see next section), or users might create derived values (see section after next) and filter based on the results.

SORT

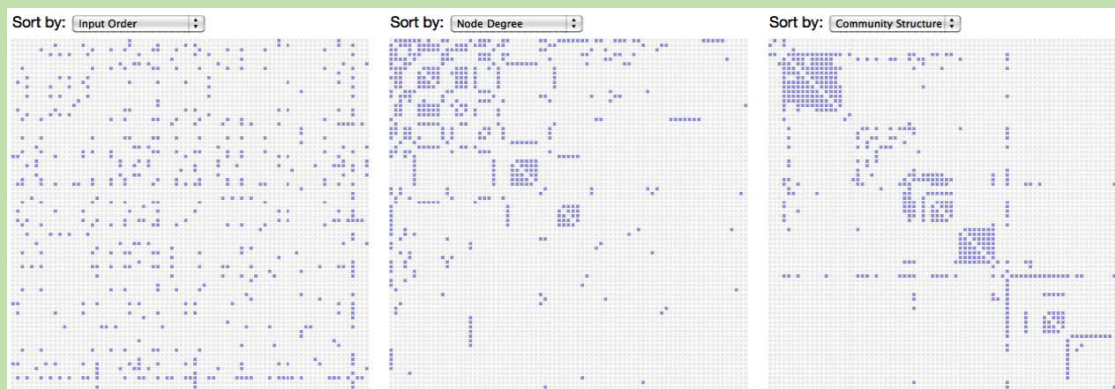
Ordering (or sorting) is another fundamental operation within a visualization. A proper ordering can effectively surface trends and clusters of values⁵ or organize the data according to a familiar unit of analysis (days of the week, financial quarters, etc.). The most common method of ordering is to sort records according to the value of one or more variables. Sorting controls can be simple choices in a toolbar or clicks on the header of a table to produce ascending or descending sorts for numerical or textual values. Sometimes specialized sort orders such as weekday or month names are necessary to reveal important patterns.

Ordering becomes more complicated in the case of multiple view displays, in which both entire plots and the values they contain may be sorted to reveal patterns or anomalies. Sorting values consistently across plots (for example, by their marginal mean or median values) can reveal patterns while facilitating comparison among plots.

Some data types (e.g., multivariate tables, networks) do not always lend themselves to simple sorting by value. Such data may require more sophisticated *seriation* methods^{20,57,58} that attempt to minimize a distance measure among items. The goal is to reveal underlying structure (e.g., clustering) within the data. An example is shown in figure 5, a matrix-based visualization of a social network. On the left, a matrix plot of a social network conveys little structure when the rows and columns (representing people) are sorted alphabetically. Interactively reordering the matrix by node degree reveals more structure (center). Seriating the matrix by network connectivity reveals underlying clusters of communities (right).

FIGURE 5

Reorderable matrices



DERIVE

As an analysis proceeds in iterative cycles, users may find that the input data is insufficient: variables may need to be transformed or new attributes derived from existing values. Common cases include normalization or log transforms to enable more effective value comparisons. Derived measures are often used to summarize the input data, ranging from descriptive statistics (mean, median, variance) to model fitting (regression curves) and data transformation (group-by aggregation such as counts or summations). While analysts can derive new values prior to importing data for visual analysis, the overhead of moving between tools stymies fluid, iterative exploration. As a result, visual analytics tools should include facilities for deriving new data from input data. Often this functionality is provided via a *calculation language*, similar to those found in spreadsheets or database query languages. Beyond these basic functions, hypothesis-testing methods (t-tests, ANOVA) can amplify the benefits of smooth integration of statistics and visualization.

Improved derivation methods present a promising frontier for visual analytics research. How can visual tools support flexible construction of more advanced models or derived values? Using programming-by-demonstration methods, analysts might annotate patterns (e.g., of network intrusion events⁶⁰) from which the system can generalize pattern-recognition rules. Or, visual tools might automatically fit applicable statistical models to the data based on the current visualization state. For example, the nesting of variables within common “pivot” displays could be mapped to the structure of a linear model. More principled frameworks that wed visualization to modeling and forecasting are still emerging.

VIEW MANIPULATION

Once analysts have created a visualization through data and view specification actions, they should be able to manipulate the view to highlight patterns, investigate hypotheses, and drill down for more details. Analysts must be able to *select* items or data regions to highlight, filter, or operate on them. Large information spaces may require analysts to scroll, pan, zoom, and otherwise *navigate* the view to examine both high-level patterns and fine-grained details. Multiple, linked visualizations often provide clearer insights into multidimensional data than do isolated views. Analysis tools must be able to *coordinate* multiple views so that selection and filtering operations apply to all displays at once and *organize* the resulting dashboards and work spaces.

SELECT

Pointing to an item or region of interest is common in everyday communication because it indicates the subject of conversation and action. In the physical world, people coordinate their gestures, gaze, and speech to indicate salient items. For example, different hand gestures can communicate angle (oriented flat hand), height (horizontal flat hand), intervals (thumb and index finger in “C” shape), groupings (circling a region), and forces (accelerating fist).²⁷ In visual analysis, reference (or *selection*) remains of critical importance, but it is realized through a more limited set of actions, such as clicking or lassoing items of interest.

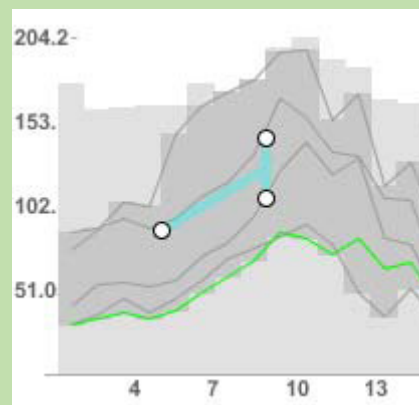
Common forms of selection within visualizations include mouse hover, mouse click, region selections (e.g., rectangular and elliptical regions, or free-form “lassos”), and area cursors (e.g., “brushes”⁴ or dynamic selectors such as the bubble cursor,¹⁸ which selects the item currently closest to the mouse pointer).

These selections often determine a set of objects to be manipulated, enabling highlighting, annotation, filtering, or details-on-demand. Note that interactive selection is closely related to filtering: selections can be used to identify items to remove from the display. The context of interaction must also be taken into account when choosing a selection method. For example, responding to hover events to provide details-on-demand is inappropriate when using touch-based input on a tablet or mobile phone.

Selections can also vary in terms of their expressive power. Most interfaces support selections of a collection of items. Though this approach is easy to implement, it does not allow analysts to specify higher-level criteria. A more powerful, albeit more complex, approach is to support selections as queries over the data.²² Maintaining query structure increases the expressiveness of visualization applications. For example, rather than directly selecting the contained items, drawing a rectangle in a chart may specify a range query over the data variables represented by the X and Y axes. The resulting selection criteria can then be saved and applied to dynamic data (updating items may enter or exit a query region) or to a completely different visualization. Examples include querying stock-price changes in TimeSearcher²⁸ (see figure 6) and attribute ranges in parallel coordinates displays³⁰ (figure 7). In figure 6 an angular selection tool specifies a target slope (rate of change) and tolerance for a collection of stock prices. All time series with a similar slope over the queried time range are selected; shaded regions show envelopes of minimum and maximum values. The widget operates directly on the visualization: dragging the widget from left to right interactively queries other time windows. In figure 7 parallel coordinates plot multidimensional data as line segments among parallel axes. Here, an analyst has dragged along the axes to create interactive selections that highlight automobiles with low weight and high mileage.

FIGURE 6

Querying time-series by slope in TimeSearcher²⁸



Designing more expressive selection methods remains an active area of research. For example, researchers have proposed methods to map mouse gestures over a time-series visualization to select perceptually salient data regions such as peaks, valleys, and slopes³⁵ (see figure 8) or to query complex patterns of temporal variation.²⁹ Initial selections can also be used as a starting point for more complex selections, as analysts might click a representative object and then formulate a broader selection based on the object's properties (e.g., “select all items blue like this one”).²² Of course, selection need not be limited to the mouse and keyboard: input modalities such as touch, gesture, and speech might enable new, effective forms of selection.

FIGURE 7

Selection queries in parallel coordinates

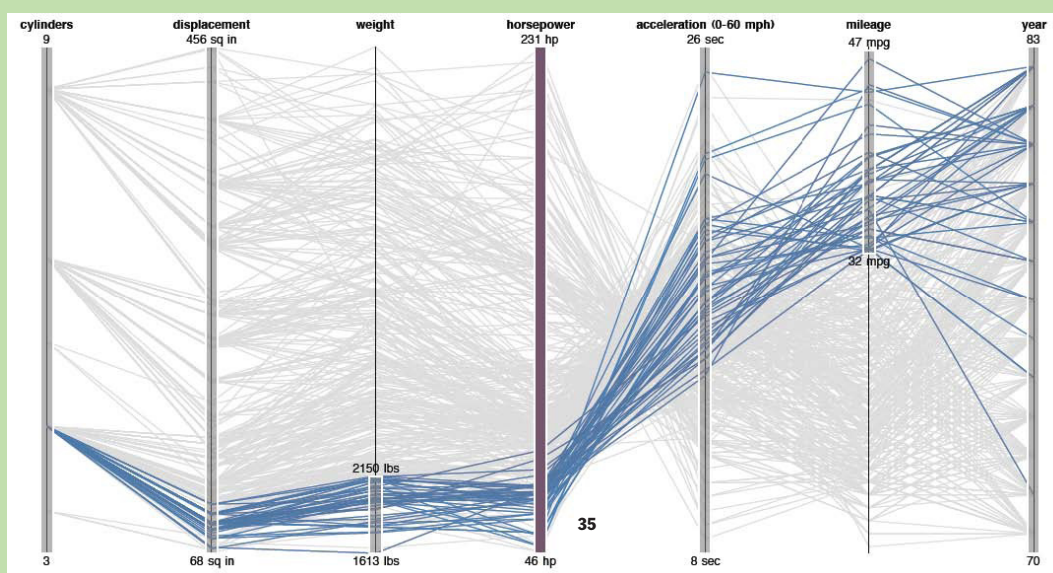
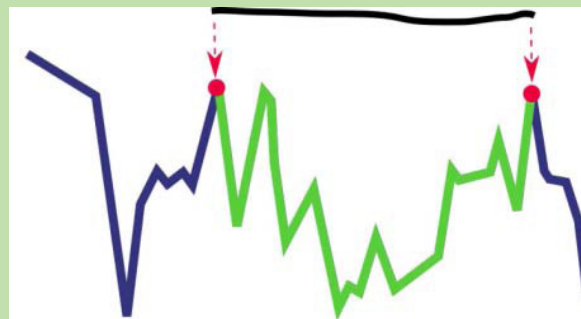


FIGURE 8

Perceptual interpretation of Ink annotations



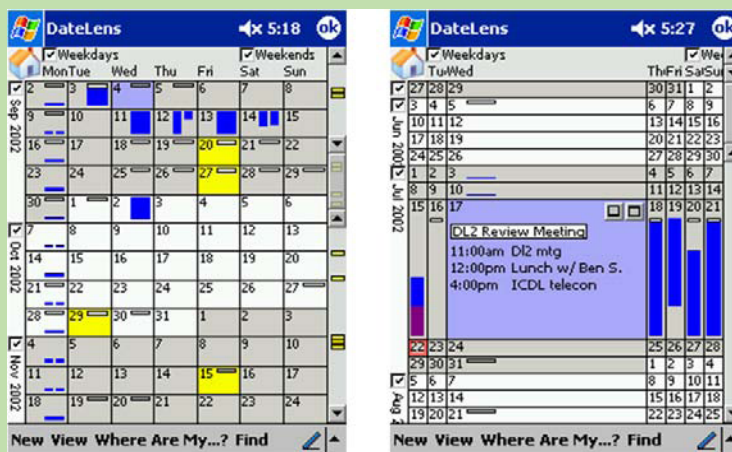
NAVIGATE

How analysts navigate a visualization is in part determined by where they start. One common pattern of navigation adheres to the widely cited visual information-seeking mantra: “Overview first, zoom and filter, then details-on-demand.”⁴⁸ Analysts may begin by taking a broad view of the data, including assessment of prominent clusters, outliers, and potential data-quality issues. These orienting actions can then be followed by more specific, detailed investigations of data subsets. A common example is geographic maps: an overview might show an overall territory, followed by zooming into regions of interest. For example, the map in figure 4 depicts criminal activity by time and region. It shows all crimes committed after dark during the last week of October 2011. Dynamic query widgets enable filtering by time of day (left), date span (bottom), and type of crime (right). Pan (drag) and zoom (buttons and scroll wheel) controls enable view navigation. As an analyst zooms in on the map, the circular crime markers gain detailed labels—a form of *semantic zooming*.

Of course, starting with an expansive overview is not always advisable. A legal analyst researching for an upcoming trial may be wise to forego an overview of the entire history of U.S. court decisions. Instead, the analyst might start with the legal decisions most relevant to the current case, perhaps determined by keyword search, and expand the investigation to other, cited decisions. This form of navigation can be summarized as “Search, show context, expand on demand.”⁵³

In either case, visualizations often function as *viewports* onto an information space. Analysts need to manipulate these viewports to navigate the space. Common examples include scrolling or panning a display via scrollbars or mouse drag, and zooming among different levels using a zoom slider or scroll wheel (figure 4). Zooming need not follow a strict geometric metaphor: semantic zooming⁷ methods can modify both the amount of information shown and how it is displayed as analysts move among levels of detail. In the calendar in figure 9, the display magnifies selected regions as analysts navigate from months to days to hours. Semantic zooming reveals more details within focal regions. Additionally, dynamic query widgets, such as range sliders for the X and Y axes

FIGURE 9

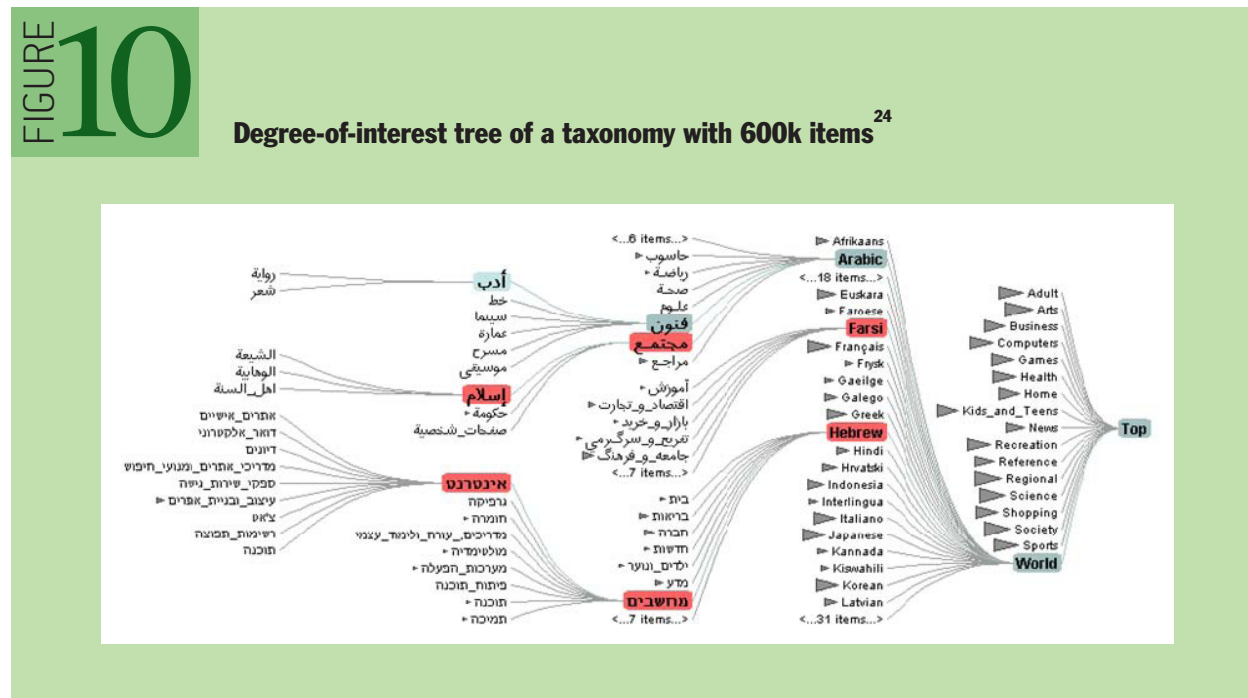
DateLens Focus-plus-Context Calendar⁶

of a scatter plot, can filter the visible data range and thus provide a form of zooming within a chart.

To aid navigation further, researchers have developed a variety of *focus plus context* methods. These “bifocal” views⁴⁹ provide a detailed view of a high-interest data region while retaining surrounding context to help keep analysts oriented. A second key idea is the use of *overview and detail* displays. For example, a geographic visualization might include a large zoomed-in map (the detail), while a smaller, zoomed-out map includes a rectangle showing the position of the zoomed-in view within the broader terrain (the overview). In this case, the detail view provides the focus, and the overview provides context. The benefits are highest when the zoom factor (ratio of overview to detail view) is 5-20.⁴² When larger zoom factors are needed, intermediate overviews may also be helpful.

A different approach is to use *distortion* or magnification techniques that transform the entire display region such that contextual regions are demagnified. A simple example is the Mac OS X dock, which uses 1D fisheye distortion to show common applications; more sophisticated methods employ distortion in multiple dimensions. While often visually intriguing, complex distortion methods have yet to prove their worth in real-world applications: viewers can become disoriented by nonlinear distortions, which show no significant performance improvement over simpler methods such as zooming.³⁹

In addition to manipulating display space, focus-plus-context methods can be applied directly to the data itself. The goal is to identify which data items are currently of high interest (focus), which are of high importance regardless of the current focus (context), and which can be safely removed from view. *DOI (degree-of-interest)* functions^{17,24,53} calculate scores for information content based both on general importance (e.g., top-level categories within a hierarchy, or nodes with high centrality in a graph) and current interest (e.g., as indicated by mouse clicks, search queries, or proximity to other high-interest items). The distribution of DOI scores can then be used to selectively control the visibility of items based on the current view size and context of interaction, as in figure 10. As analysts click on or search for different items, the DOI scores



dynamically update to reveal relevant unseen data or hide irrelevant detail. A model of the analyst's current interest filters the display to the most relevant items. Low-interest items are elided but still accessible through aggregate representations. The interest estimates update as an analyst explores the taxonomy, initiating animated transitions between different views of the data.

Visualizations can provide cues to assist analysts' decisions of where and how to navigate. The controls for view manipulation have often been invisible, such as zooming/panning by mouse movement. Improved strategies facilitate discovery by analysts and provide visible indication of settings in legends or other ways, such as scrollbar positions, that provide informative feedback. An important challenge is to show selected items, even when they are not in view. For example, the results of a text search that are not currently in view might be shown by markers in the scrollbar⁶¹ or the periphery of the display.^{3,19}

COORDINATE

Many analysis problems require *coordinated multiple views* that enable analysts to see their data from different perspectives. A public policy analyst studying educational attainment might produce a bar chart of people's ages, a map of locations, a textual list with education history, and a scatter plot showing income vs. education. By selecting a single item or a group in one view, analysts might see related details or highlighted items in the other views. This powerful approach to exploring multivariate data also enables drilling down into subgroups, marking sets, and exporting selections.

Multiview displays can facilitate comparison. For example, Edward Tufte⁵² advocates the use of *small multiples*: a collection of visualizations placed in spatial proximity and typically using the same measures and scales. As in figure 11, these small multiples, also called *trellis plots*, enable rapid comparison of different data dimensions or time slices. The visualization shows employment figures by economic sector in Minnesota. The repetition of the chart form supports comparison among sectors. Plotting all the data in one chart would otherwise clutter and obscure individual trends. Selecting a point in time in one view highlights the corresponding point in all other views.

FIGURE 11

Small multiples display



Alternatively, multiple view displays can use a variety of visualization types— such as histograms, scatter plots, maps, or network diagrams— to show different projections of a multidimensional data set. An analyst constructs a complex patchwork of interlinked tables, plots, and maps in figure 12 to analyze the outcomes of elections in Michigan. Annotations indicate how selected data items correspond between visualization views. Accompanying items such as legends, histogram sliders, and scrollbars with highlighting markers can all provide multiple views onto the data. Automatically generated legends and axes are important for providing accurate annotations for analysts and meaningful explanations when visualizations are shared. Legends and axes can also become control panels for changing color palettes, marker attributes, variable ranges, or provenance information.⁴³

Multiview displays can also enable interactive exploration across views. *Brushing and linking* is the process of selecting (brushing) items in one display to highlight (or hide) corresponding data in the other views.⁴ In figure 13, a baseball analyst makes selections in one plot and corresponding items highlight in the others. On the left, selecting high-income players (top-right plot) shows little dependence on career length or fielding ability, but correlates with hitting performance. On the right, selecting the cluster of players who make more assists than put-outs (middle-left plot) reveals a strong dependence on position. Each visualization can thus serve as an input channel for revealing patterns across a data set. Linked selection enables rich, multidimensional reasoning by allowing analysts to assess how patterns in one view project onto the others. Analysts may wish to coordinate views in variety of ways:^{40,55} selecting items in one view might highlight matching records in other views, or instead provide filtering criteria to remove information from the other displays. Linked navigation provides an additional form of coordination: scrolling or zooming one view can simultaneously manipulate other views.

Though comparing multiple visualizations requires viewers to orchestrate their attention and mentally integrate patterns among views, this process is often more effective than cluttering a single visualization with too many dimensions. Future studies of how analysts construct multiview

FIGURE 12

Multiple coordinated views in *Improvise*⁵⁵

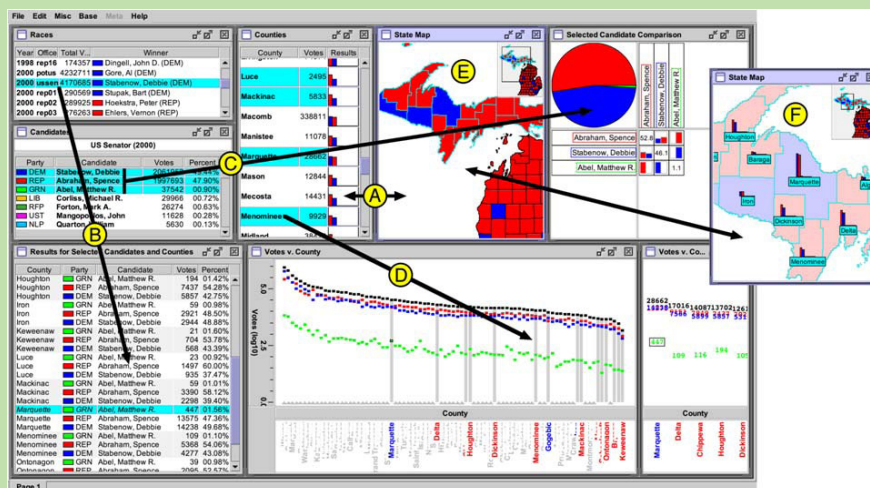
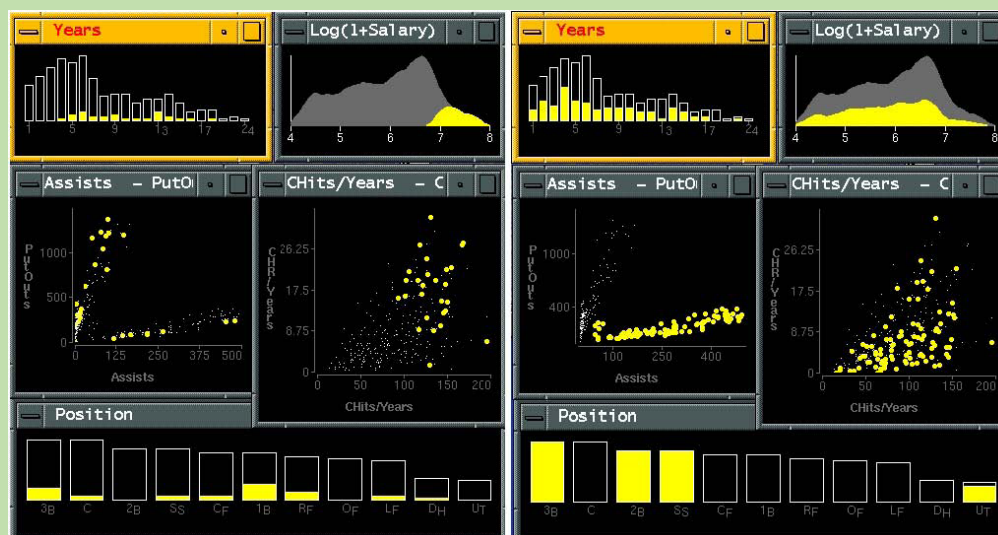


FIGURE 13

Brushing and linking of baseball statistics in GGobi



displays and specify coordination behaviors (e.g., highlighting, filtering) could provide designers with an understanding of how to build more effective tools. In addition, if designers ensure that rich multiview displays stay understandable, analysts are more likely to make compelling insights. Newcomers to an analysis, or even seasoned analysts simply returning from a coffee break, may become confused by the number of views and the potentially complicated set of coordinated queries between them. Visual analytics systems that provide access to coordination settings and replay the history of view construction can enhance understanding.

ORGANIZE

When analysts make use of multiple views they face the corresponding challenge of managing a collection of visualizations. As in traditional window-based interfaces, analysts may wish to open, close, maximize, and lay out different components. As purely manual window manipulation can be tedious, well-designed visual analytics tools simplify the organization of visualization views, legends, and interactive controls. For example, a human resources data set may show a scatter plot of salary by years of experience, plus a bar chart showing 10 age groups, and a treemap with seven corporate sites, each with 10-30 job titles. These three visualizations might give a large area for the scatter plot, with the bar chart and treemap to the right side stacked one above the other. A control panel with sliders, checkboxes, radio buttons, and a search box could be on the far right, with a details-on-demand window and annotation box across the bottom. This tiled approach allows analysts with sufficiently large displays to see all the information and selectors at once, minimizing distracting scrolling or window operations, while enabling them to concentrate on extracting and reporting insights. The coordination across windows means that slider movements or checkbox selections will

cause all views to update, allowing rapid exploration of just the employees at certain sites or specific job titles.

Typical systems allow analysts to add views, such as a second scatter plot, in ways that make modest changes to the existing window organization. An alternative approach is to add a new tab that contains the second scatter plot, so analysts can switch between the first and second set of windows. A common feature is to add trellised views, so multiple visualizations can be created at once— for example, separate bar charts showing age distributions for each of the seven corporate sites.

More advanced systems might aid this process through automated support⁸ that enables multiple windows to be opened/closed as a group and lays them out in orderly ways. Useful methods include standard scatter plot matrices (showing all pairs of scatter plots) or custom generation of related views of interest (e.g., of data variables correlated to the visualized attributes). Desirable features are automatic (re)sizing as views are added or removed and layout routines to place related views in spatial proximity.

As larger and multiple displays become more common, layout organization tools will become decisive factors in creating effective user experiences. Similarly, the demand for tablet and smartphone visualizations will promote innovation in layout organizations that are compact and reconfigurable by simple gestures. Zooming, panning, flipping, and sequencing strategies will also improve analyst experiences and facilitate effective presentations to others.

PROCESS AND PROVENANCE

Visual analytics is not limited to the generation and manipulation of visualizations— it involves a process of iterative data exploration and interpretation. As a result, visual analytics tools that provide facilities for scaffolding the analysis process will be more widely adopted. Tools should preserve analytic provenance by keeping a *record* of analyst actions and insights so that the history of work can be reviewed and refined. Textual logs of activity have benefits, but visual overviews of activity can be more compact and comprehensible. If analysts can *annotate* patterns, outliers, and views of interest, they can document their observations, questions, and hypotheses. In a networked environment, analysts should be empowered to *share* results and discuss with colleagues, coordinate the work of multiple groups, or support processes that may take weeks and months. Moreover, analysis tools can explicitly *guide* novices through common analysis tasks, provide progress indicators for experts, or lead viewers through an analysis story.

RECORD

When analyzing data with visualizations, users regularly traverse the space of views in an iterative fashion. Exploratory analysis may result in a number of hypotheses, leading to multiple rounds of questions and answers. Analysts can generate unexpected questions that may be investigated immediately or revisited later. After conducting analysis, analysts may need to review, summarize, and communicate their findings, often in the form of reports or presentations.

To support iterative analysis, visual analysis tools can record and visualize analysts' *interaction histories*. At a minimum, applications should provide basic undo and redo support. While low-level input such as mouse and keyboard events are easy to capture, histories become much more valuable when they record high-level semantic actions. By modeling the space of user actions (view

specifications, sorting, filtering, zooming, etc.), richer logs can be constructed and visualized.^{15,25,32,45} Common visual representations of analytic actions include both chronological (“timeline”) and sequential (“comic strip”) views. As shown in figure 14, a “comic strip” display retraces the steps taken in a visual analysis of business operations data.

Visual histories also reveal the hierarchical patterns of branching histories. Reading the graph shown in figure 15 in a snake-like fashion (first left-to-right, then right-to-left) reveals patterns of iterative exploration, branching, and backtracking in an analysis. Techniques for “chunking” related actions together can further reduce clutter.^{25,36}

Visual histories can support a range of interactions. First, histories provide a convenient mechanism to revisit prior analysis states and resume incomplete explorations. Adding metadata such as comments, tags, or ratings to states can facilitate later review and sharing. Interactive histories can also capture a repeatable sequence of operations that can be named and saved as a

FIGURE 14 Visual analysis history²⁵

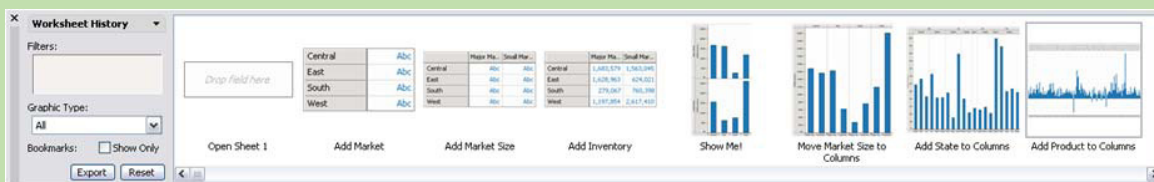
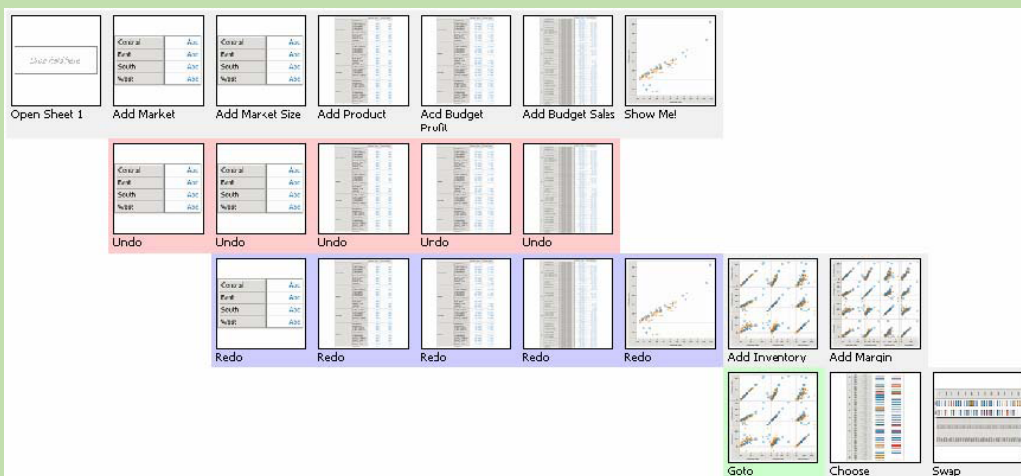


FIGURE 15 Behavior graph of branching analysis²⁵



reusable macro. This powerful feature enables analysts who are dealing with many similar data sets to automate their efforts. Histories might spur sharing: analysts can export selected analysis trails, ranging from screen shots to interactive presentations, to external media. Finally, histories also provide a means to study analysts and model analytic processes.^{32,45}

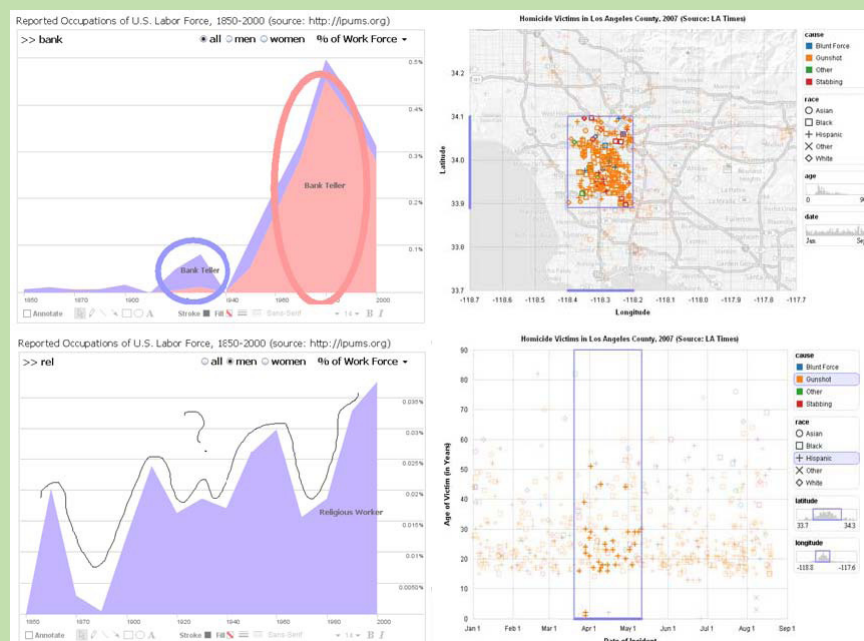
ANNOTATE

Interactive visualizations often serve not only as data exploration tools, but also as a means for recording, organizing, and communicating insights gained during exploration. One option is to allow textual annotation of states within a visual history. More expressive annotations are possible through direct interaction with the view, using the selection techniques discussed earlier. Analysts may wish to “point” to specific items or regions within a visualization and associate these annotations with explanatory text or links to other views.²⁶

Freeform *graphical annotations* provide one expressive form of pointing.²⁶ Drawing a circle around a cluster of items or pointing an arrow at a peak in a graph can direct the attention of viewers. The angle or color of the arrow or shape of the hand-drawn circle may communicate emotional cues or add emphasis. The left side of figure 16 shows annotated occupational data: the top annotation highlights a gender reversal among bank tellers using color-coded ellipses; the bottom annotation expresses confusion regarding the erratic percentage of religious workers. Although such drawings allow a high degree of expression, they lack an explicit tie to the underlying data. Freeform annotations implemented as vector graphics can persist over geometric transformations such as panning and zooming, but if they are not “*data-aware*,” then they may become meaningless in the

FIGURE 16

Annotated visualizations



face of operations such as filtering or aggregation.

Annotations can be made data-aware when realized as selections, as seen in the right side of figure 16. In the top chart, selection queries anchor annotations of crime data. The bottom chart shows annotations transferred across a change in visual encodings: the selected geographic range is now conveyed using histogram sliders. These selections can be represented as a set of selected items, a declarative query, or both.²² Data-aware annotations allow a pointing intention to be reapplied to different views of the same data, enabling reuse of references across different choices of visual encodings. Data-aware annotations may also enable analysts to search for all commentary or visualizations that reference a particular data item. As data-aware annotations are machine readable, they might also be used to export selected data or aggregated to identify data subsets of high interest.

SHARE

Researchers in visual analytics often focus on the perceptual and cognitive processes of a single analyst. In practice, real-world analysis is also a social process that may involve multiple interpretations, discussion, and dissemination of results.^{26,54} The implication is clear: to support the analysis life cycle fully, visual analytics tools should support social interaction. At minimum, tools must be able to export views (png, jpg, ppt, etc.) or data subsets (csv, json, xls, etc.) for sharing and revisitation. An important capability is to export the settings for the control panels, so other analysts can see the same visualization. Figure 17 shows sense.us²⁶ one example of a collaborative visual analysis tool incorporating view sharing, annotation, and discussion. The system consists of (a) an interactive visualization, (b) a set of graphical annotation tools, (c) bookmark trails for saved views, (d) a text-entry field for adding comments (bookmarks can be dragged onto the text field to link views to a comment), (e) textual comments attached to the current view, and (f) a shareable URL that is updated automatically as the visualization state changes.

A simple but effective aid to collaboration is view sharing via *application bookmarking*: a visual analytics system should be able to model and export its internal state.^{26,54} Unlike a static screen shot, bookmarking enables analysts to take up an exploration where their collaborators left off. View sharing often takes the form of an URL or similar identifier that allows a collaborator to navigate quickly to a view of interest. Seeing an identical view provides collaborators with a common ground for discussion. Annotation methods can be applied within such views to further collaboration. One challenge for effective view sharing concerns how to handle dynamic data: should a bookmarked view maintain a snapshot to historical data, provide access to the most current data, or both?

Another method of sharing and dissemination is to *publish* a visualization. Commercial tools such as Spotfire and Tableau can publish visualization dashboards as interactive Web pages. These Web-based components provide a subset of interactive functionality (e.g., selection, search, and drill-down) to enable some amount of follow-up analysis. Services such as IBM's Many Eyes⁵⁴ can be used to embed visualization applets in external Web sites. Publishing is particularly important for reaching larger audiences. While publishing is a necessary condition for broad sharing, it may not be sufficient by itself for engaging viewers.²¹ Visualizations embedded within a blog or discussion forum can reach an established audience and may foster discussion more effectively than a centralized site.¹³

Other collaborative concerns depend on the context of use. Are collaborators working *synchronously* (same time) or *asynchronously* (different time)? Are they *co-located* (same place) or

distributed (different place)? Each of these configurations may require specialized strategies that consider the division of work, access control, presence indicators, and activity awareness.^{21,26,31}

GUIDE

The exploration process is well understood for some traditional domains. For example, a very simple workflow might remove incomplete data items, sort, select high-value items, and report on these selections. Analysts, however, may need to develop new strategies that are formalized to guide newcomers and provide progress indicators to experts. Visual-analysis systems can incorporate *guided* analytics to lead analysts through workflows for common tasks.

Some processes are clearly linear, but many visual-analytics tasks require richer *systematic yet flexible* processes that allow analysts to take excursions while keeping track of what they have done. For example, SocialAction⁴¹ organizes social-network analysis into a sequence of activities (e.g., rank nodes, plot nodes, find communities); the system allows analysts to skip steps selectively and keeps a record of which steps have been completed. In figure 18, the panel on the left suggests common steps to structure social network analysis and provides progress indicators.

In a related vein, experts often develop visualizations that are used by less knowledgeable team members, in much the same way that spreadsheet macros enable specialists to encode accounting or business practices for others. More research is needed to identify effective visual-analytics processes and enable expert analysts to create reusable workflows.

In recent years, journalists have been experimenting with different forms of *narrative* visualization⁴⁶ by structuring interactive graphics to tell stories with data. Visualizations from *The*

FIGURE 17

Collaborative visual analysis in Sense.us⁵⁶

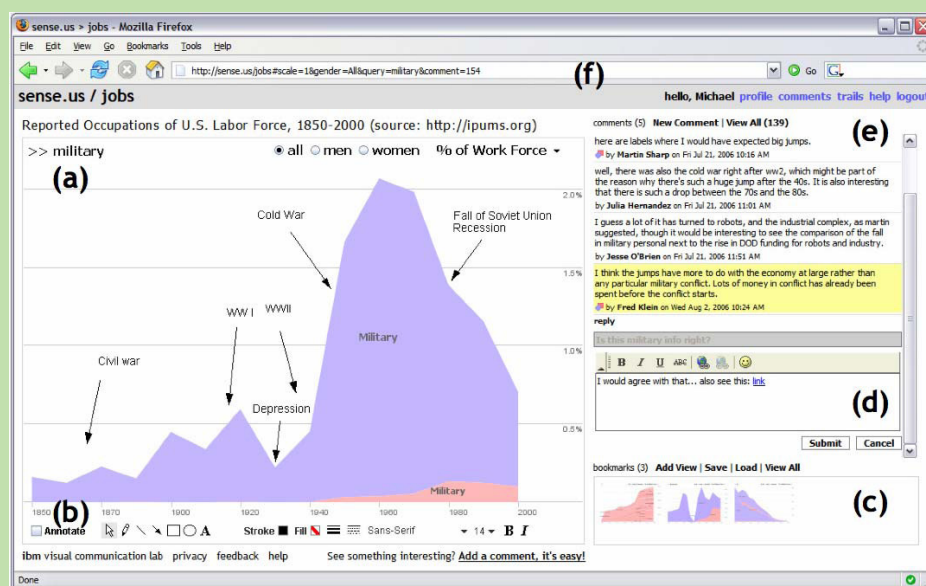
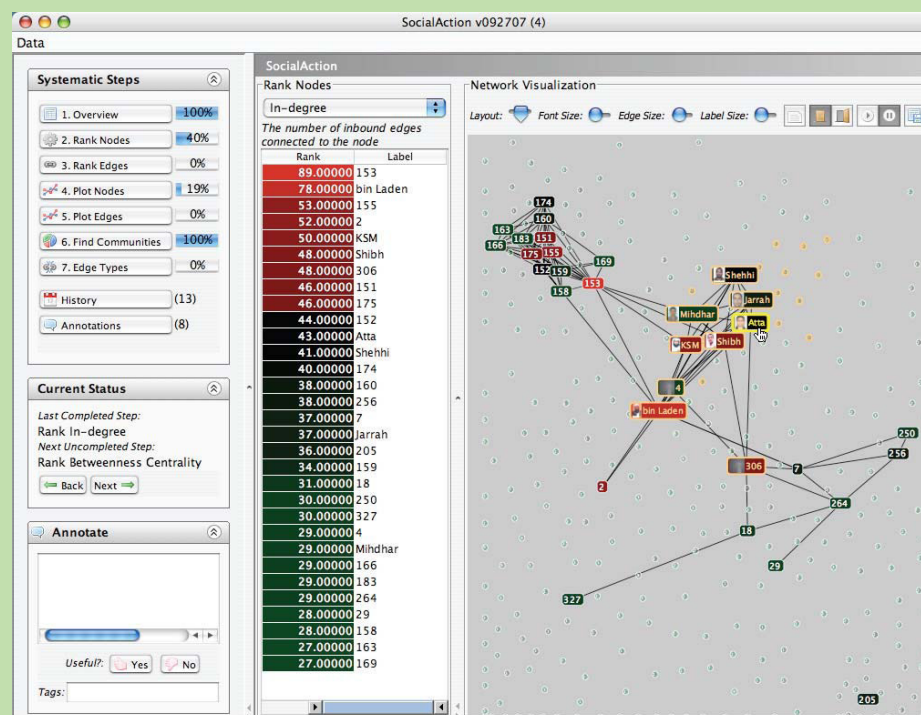


FIGURE 18

Systematic yet flexible analysis in SocialAction⁴¹

New York Times, *The Washington Post*, *The Guardian*, and other news sources often lead the viewer step by step through a linear narrative, guided by supporting text and annotations. In figure 19, for example, an interactive graphic uses staging and annotation to guide the reader through decades of budget predictions. At a story's conclusion, such visualizations provide interactive controls for further exploration. These narrative structures both communicate key observations from the data and cleverly provide a *tacit tutorial* of the available interactions by animating each component along with the story. By the time the presentation opens up for freeform exploration, the viewers have already seen demonstrations of the interactive controls. These and other forms of narrative visualization demonstrate how guided analytics can be used to disseminate data-driven stories to a general audience.

CONCLUSION

We hope this taxonomy and discussion will help advance visual analytics on multiple fronts. For students and newcomers to the field, the taxonomy provides an orienting, high-level introduction to the interactive concerns at the heart of successful visual analysis. We encourage interested readers to consult the systems, books, and research papers referenced in this article to develop a deeper understanding of these concerns. For developers, the taxonomy can function as a checklist of

items to consider when creating new analysis tools. For researchers, the taxonomy helps highlight critical areas that would benefit from further investigation, including new methods for interactive view specification, a closer integration of visualization and statistical algorithms, selection and annotation techniques that leverage data semantics, and effective approaches to guided analytics.

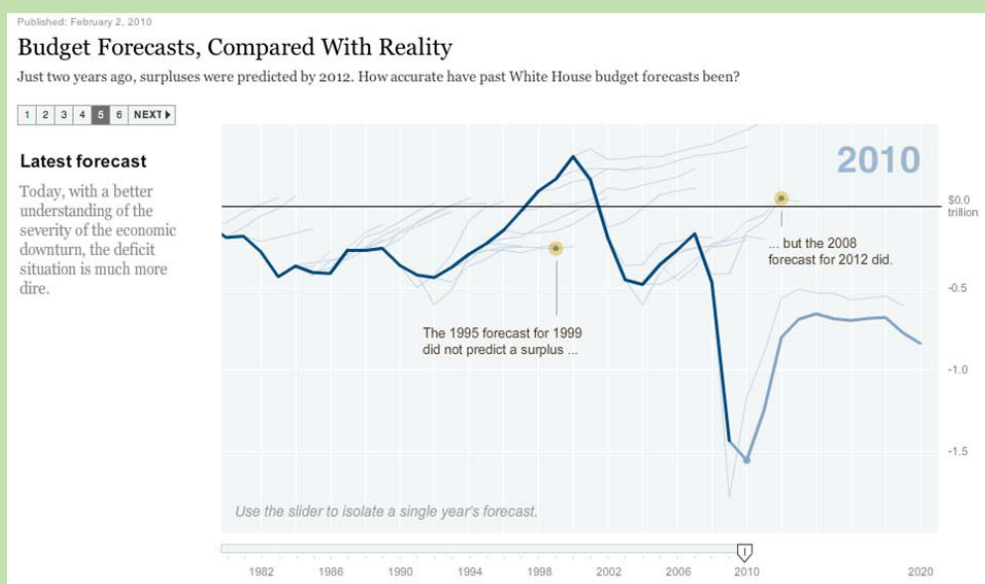
Of course, by attempting to provide an abstracted picture of a domain, taxonomies may be incomplete. In some cases, we separately categorize aspects that are closely related. Dynamic query widgets enabling data specification often serve as a means of view navigation. Selection techniques are also central to effective annotation schemes.

In other instances, we selectively omit material. First, we do not go into great depth regarding implementation details. Supporting realtime interactivity often requires careful attention to system design, especially for large data sets. While popular platforms for large data analysis such as MapReduce¹⁴ achieve adequate throughput, their high latency and lack of online processing limit fluent interaction. The demands of truly interactive analysis pose important research challenges for the designers of analysis platforms, ranging from low-latency architectures to intelligent sampling and aggregation methods.³⁴

Our taxonomy is also somewhat sparing in its discussion of the current frontier of visual analytics research. For example, how best to incorporate (semi-)automated statistical methods within a visualization environment is a central challenge. Our discussion of derived data only scratches the surface. A related concern is the task of *data wrangling*:³³ reformatting, cleaning, and integrating data sets so that they are amenable to visual analysis. Incorrect or improperly structured data diverts the attention and energy of trained analysts and presents a significant barrier to newcomers. As data cleaning requires nuanced human judgment based on domain knowledge (“is this outlier an error

FIGURE 19

Data storytelling by The New York Times



or a discovery?”), data wrangling is a necessarily interactive process combining statistical methods, visualization, and interaction techniques. This topic deserves a deeper treatment than we can provide within our compact taxonomy.

These concerns represent active areas of research, and we expect our characterization of the field to evolve in the years to come. Validating and evolving this framework is a community project that can proceed through feedback, critique, and refinement by visual analytics researchers and practitioners. We invite the insights and commentary of the visualization, statistics, database, and HCI (human-computer interaction) communities, and eagerly anticipate the continued flowering of improved tools for making sense of the wealth of data that surrounds us.

ACKNOWLEDGMENTS

We thank our colleagues and students for providing valuable comments on drafts: Maneesh Agrawala, Jason Chuang, Cody Dunne, John Guerra-Gomez, Pat Hanrahan, Sean Kandel, Diana MacLean, and Kostas Pantazos. This work was partially supported by National Science Foundation grants IIS-0968521, IIS-1017745, CCF-0964173 and SBE-0915645, NIH-National Cancer Institute grant RC1-CA147489, and ONC-SHARP grant on Cognitive Information Design and Visualization.

REFERENCES

1. Abram, G., Treinish, L. 1995. An extended data-flow architecture for data analysis and visualization. *Proceedings of the IEEE Conference on Visualization*: 263-270.
2. Ahlberg, C., Shneiderman, B. 1994. Visual information seeking: tight coupling of dynamic query filters with starfield displays. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 313-317.
3. Baudisch, P., Rosenholtz, R. 2003. Halo: a technique for visualizing off-screen objects. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 481-488; <http://doi.acm.org/10.1145/642611.642695>.
4. Becker, R. A., Cleveland, W. S. 1987. Brushing scatterplots. *Technometrics* 29(2): 127-142.
5. Becker, R. A., Cleveland, W. S., Shyu, M.-J. 1996. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics* 5(2): 123-155.
6. Bederson, B. B., Clamage, A., Czerwinski, M. P., Robertson, G. G. 2004. DateLens: a fisheye calendar interface for PDAs. *ACM Transactions on Computer-Human Interaction* 11(1): 90-119.
7. Bederson, B. B., Hollan, J. D. 1994. Pad++: a zooming graphical interface for exploring alternate interface physics. *Proceedings of the ACM Symposium on User Interface Software and Technology*: 17-26; <http://doi.acm.org/10.1145/192426.192435>.
8. Bell, B. A., Feiner, S. K. 2000. Dynamic space management for user interfaces. *Proceedings of the ACM Symposium on User Interface Software and Technology*: 239-248; <http://doi.acm.org/10.1145/354401.354790>.
9. Bestiario Impure; <http://www.impure.com/>.
10. Bostock, M., Heer, J. 2009. Protovis: a graphical toolkit for visualization. *IEEE Transactions on Visualization and Computer Graphics* 15(6): 1121-1128.
11. Card, S. K., Mackinlay, J., Shneiderman, B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.
12. Cleveland, W. S. 1994. *The Elements of Graphing Data*. Lafayette, IN: Hobart Press.
13. Danis, C. M., Viegas, F. B., Wattenberg, M., Kriss, J. 2008. Your place or mine?: visualization as a

- community component. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 275-284; <http://doi.acm.org/10.1145/1357054.1357102>.
14. Dean, J., Ghemawat, S. 2004. MapReduce: simplified data processing on large clusters. *Operating Systems Design and Implementation (OSDI)*: 137-150.
 15. Derthick, M., Roth, S. F. 2001. Enhancing data exploration with a branching history of user operations. *Knowledge Based Systems* 14(1-2): 65-74.
 16. Few, S. 2009. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Berkeley, CA: Analytics Press.
 17. Furnas, G. W. 2006. A fisheye follow-up: further reflections on focus + context. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 999-1008; <http://doi.acm.org/10.1145/1124772.1124921>.
 18. Grossman, T., Balakrishnan, R. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 281-290; <http://doi.acm.org/10.1145/1054972.1055012>.
 19. Gustafson, S., Baudisch, P., Gutwin, C., Irani, P. 2008. Wedge: clutter-free visualization of off-screen locations. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 787-796; <http://doi.acm.org/10.1145/1357054.1357179>.
 20. Hahsler, M., Hornik, K., Buchta, C. 2008. Getting things in order: an introduction to the R Package seriation. *Journal of Statistical Software* 25 (3): 1-34; <http://www.jstatsoft.org/v25/i03>.
 21. Heer, J., Agrawala, M. 2008. Design considerations for collaborative visual analytics. *Information Visualization* 7(1): 49-62.
 22. Heer, J., Agrawala, M., Willett, W. 2008. Generalized selection via interactive query relaxation. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 959-968.
 23. Heer, J., Bostock, M., Ogievetsky, V. 2010. A tour through the visualization zoo. *Communications of the ACM* 53(6): 59-67; <http://doi.acm.org/10.1145/1743546.1743567>.
 24. Heer, J., Card, S. K. 2004. DOITrees revisited: scalable, space-constrained visualization of hierarchical data. *Proceedings of Advanced Visual Interfaces*: 421-424; <http://doi.acm.org/10.1145/989863.989941>.
 25. Heer, J., Mackinlay, J., Stolte, C., Agrawala, M. 2008. Graphical histories for visualization: supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics* 14(6): 1189-1196; <http://portal.acm.org/citation.cfm?id=1477066.1477414>.
 26. Heer, J., Viégas, F. B., Wattenberg, M. 2009. Voyager and voyeurs: supporting asynchronous collaborative information visualization. *Communications of the ACM* 52(1): 87-97; <http://doi.acm.org/10.1145/1435417.1435439>.
 27. Hill, W. C., Hollan, J. D., 1991. Deixis and the future of visualization excellence. *Proceedings of the IEEE Conference on Visualization*: 314-320; <http://portal.acm.org/citation.cfm?id=949607.949659>.
 28. Hochheiser, H., Shneiderman, B. 2004. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization* 3(1): 1-18.
 29. Holz, C., Feiner, S. 2009. Relaxed selection techniques for querying time-series graphs. *Proceedings of the ACM Symposium on User Interface Software and Technology*: 213-222; <http://doi.acm.org/10.1145/1622176.1622217>.
 30. Inselberg, A., Dimsdale, B. 1990. Parallel coordinates: a tool for visualizing multi-dimensional geometry. *Proceedings of the IEEE Conference on Visualization*: 361-378.

31. Isenberg, P., Tang, A., Carpendale, S. 2008. An exploratory study of visual information analysis. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 1217-1226; <http://doi.acm.org/10.1145/1357054.1357245>.
32. Jankun-Kelly, T. J., Ma, K.-L., Gertz, M. 2007. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics* 13(2): 357-369; <http://dx.doi.org/10.1109/TVCG.2007.28>.
33. Kandel, S., Heer, J., C. Plaisant, C., Kennedy, J., van Ham, F., Henry-Riche, N., Weaver, C., Lee, B., Brodbeck, D., Buono, P. 2011. Research directions for data wrangling: visualizations and transformations for usable and credible data. *Information Visualization*.
34. Keim, D. A., Mansmann, F., Schneidewind, J., Ziegler, H. 2006. Challenges in visual data analysis. *Information Visualization*: 9-16.
35. Kong, N., Agrawala, M. 2009. Perceptual interpretation of ink annotations on line charts. *Proceedings of the ACM Symposium on User Interface Software and Technology*: 233-236; <http://doi.acm.org/10.1145/1622176.1622219>.
36. Kurlander, D., Feiner, S. 1988. Editable graphical histories. *Proceedings of the IEEE Workshop on Visual Language*: 127-134.
37. Mackinlay, J. D. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics* 5(2): 110-141; <http://doi.acm.org/10.1145/22949.22950>.
38. Mackinlay, J. D., Hanrahan, P., Stolte, C. 2007. Show me: automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics* 6: 1137-1144.
39. Nekrasovski, D., Bodnar, A., McGrenere, J., Guimbreti re, F., Munzner, T. 2006. An evaluation of pan and zoom and rubber sheet navigation with and without an overview. *Proceedings of the ACM Conference on Human Factors in Computing Systems*: 11-20; <http://doi.acm.org/10.1145/1124772.1124775>.
40. North, C., Shneiderman, B. 2000. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. *Proceedings of Advanced Visual Interfaces*: 128-135; <http://doi.acm.org/10.1145/345513.345282>.
41. Perer, A., Shneiderman, B. 2008. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. *Proceedings of Intelligent User Interfaces*: 109-118; <http://doi.acm.org/10.1145/1378773.1378788>.
42. Plaisant, C., Carr, C., Shneiderman, B. 1995. Image-browser taxonomy and guidelines for designers. *IEEE Software* 12(2): 21-32.
43. Riche, N. H., Lee, B., Plaisant, C. 2010. Understanding interactive legends: a comparative evaluation with standard widgets. *Computer Graphics Forum* 29(3): 1193-1202.
44. Roth, S. F., Mattis, J. 1991. Automating the presentation of information. *Proceedings of the IEEE Conference on Artificial Intelligence Applications*: 90-97.
45. Scheidegger, C., Koop, D., Santos, E., Vo, H., Callahan, S., Freire, J., Silva, C. 2008. Tackling the Provenance Challenge One Layer at a Time. *Concurrency and Computation: Practice and Experience* 20(5): 473-483; <http://portal.acm.org/citation.cfm?id=1350745.1350757>.
46. Segel, E., Heer, J. 2010. Narrative visualization: telling stories with data. *IEEE Transactions on Visualization and Computer Graphics* 16(6): 1139-1148.

47. Shneiderman, B. 1994. Dynamic queries for visual information seeking. *IEEE Software* 11(6): 70-77.
48. Shneiderman, B. 1996. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*; <http://portal.acm.org/citation.cfm?id=832277.834354>.
49. Spence, R., Apperley, M. 1982. Data base navigation: an office environment for the professional. *Behaviour and Information Technology* 1(1): 43-54.
50. Stolte, C. Tang, D., Hanrahan, P. 2002. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics* 8: 52-65.
51. Tableau Software; <http://tableausoftware.com>.
52. Tufte, E. 1983. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.
53. van Ham, F., Perer, A. 2009. Search, show context, expand on demand: supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics* 15(6): 953-960; <http://dx.doi.org/10.1109/TVCG.2009.108>.
54. Viégas, F. B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M. 2007. Many Eyes: a site for visualization at Internet scale. *IEEE Transactions on Visualization and Computer Graphics* 13(6): 1121-1128.
55. Weaver, C. E.. 2004. Building highly-coordinated visualizations in Improvise. *Proceedings of the IEEE Information Visualization Conference*: 159-166.
56. Wickham, H. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer.
57. Wilkinson, L. 2005. *The Grammar of Graphics* (Statistics and Computing). Secaucus, NJ: Springer-Verlag.
58. Wilkinson, L., Friendly, M. 2009. The history of the cluster heat map. *The American Statistician* 63(2): 179-184.
59. Willett, W., Heer, J., Agrawala, M. 2007. Scented widgets: improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13(6): 1129-1136; <http://dx.doi.org/10.1109/TVCG.2007.70589>.
60. Xiao, L., Gerth, J., Hanrahan, P. 2006. Enhancing visual analysis of network traffic using a knowledge representation. *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*: 107-114.
61. Zellweger, P. T., Mackinlay, J. D., Good, L., Stefik, M., Baudisch, P. 2003. City lights: contextual views in minimal space. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*: 838-839; <http://doi.acm.org/10.1145/765891.766022>.

LOVE IT, HATE IT? LET US KNOW

feedback@queue.acm.org

JEFFREY HEER (jheer@cs.stanford.edu) is an assistant professor of computer science at Stanford University, where he works on human-computer interaction, visualization, and social computing. In 2009, he was named to *MIT Technology Review's* TR35 (35 innovators under 35). He holds B.S., M.S., and Ph.D. degrees in computer science from the University of California, Berkeley. **BEN SHNEIDERMAN** (ben@cs.umd.edu) is a professor in the department of computer science, founding director of the Human-Computer Interaction Laboratory, and a member of the Institute for Advanced Computer Studies at the

University of Maryland, College Park. He was elected as a Fellow of the ACM in 1997 and a Fellow of the American Association for the Advancement of Science in 2001. He received the ACM SIGCHI Lifetime Achievement Award in 2001. He is a member of the National Academy of Engineering.

© 2012 ACM 1542-7730/12/0200 \$10.00